

The image displays a 10x10 grid of characters forming a stylized 'S' shape. The shape is composed of 'S' and 'Y' characters. The 'S' characters form the main body of the shape, while 'Y' characters form the internal structure and the rightmost vertical bar. The shape is symmetrical about a vertical axis.

```
000000  PPPPPPP  EEEEEEEEE  NN      NN  MM      MM  SSSSSSS  GGGGGGG
000000  PPPPPPP  EEEEEEEEE  NN      NN  MM      MM  SSSSSSS  GGGGGGG
00      00  PP      PP  EE      EE  NN      NN  MM      MM  SS      SS  GG      GG
00      00  PP      PP  EE      EE  NN      NN  MM      MM  SS      SS  GG      GG
00      00  PP      PP  EE      EE  NN      NN  MM      MM  SS      SS  GG      GG
00      00  PP      PP  EE      EE  NN      NN  MM      MM  SS      SS  GG      GG
00      00  PPPPPPP  EEEEEEEEE  NN      NN  NN      NN  MM      MM  SSSSSS  GG      GG
00      00  PPPPPPP  EEEEEEEEE  NN      NN  NN      NN  MM      MM  SSSSSS  GG      GG
00      00  PP      PP  EE      EE  NN      NN  NN      NN  MM      MM  SS      SS  GG      GG
00      00  PP      PP  EE      EE  NN      NN  NN      NN  MM      MM  SS      SS  GG      GG
00      00  PP      PP  EE      EE  NN      NN  NN      NN  MM      MM  SS      SS  GG      GG
00      00  PP      PP  EE      EE  NN      NN  NN      NN  MM      MM  SS      SS  GG      GG
000000  PP      PP  EEEEEEEEE  NN      NN  NN      NN  MM      MM  SSSSSSS  GG      GG
000000  PP      PP  EEEEEEEEE  NN      NN  NN      NN  MM      MM  SSSSSSS  GG      GG
                                ....
                                ....
                                ....
                                ....
```

```
LL      IIIIII  SSSSSSS
LL      IIIIII  SSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLL  IIIIII  SSSSSSS
LLLLLLLL  IIIIII  SSSSSSS
```

(2)	45	DATA DEFINITIONS
(3)	81	OPEN MESSAGE FILES FOR SYSTEM OUTPUT AND ERROR
(4)	227	INTERNAL SUBROUTINES
(6)	299	CLOSE MESSAGE FILES

```
0000 1 .TITLE OPENMSG - OPEN MESSAGE FILES FOR SYSTEM OUTPUT AND ERROR
0000 2 .IDENT 'V04-000'
0000 3
0000 4
0000 5 *****
0000 6
0000 7 COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 ALL RIGHTS RESERVED.
0000 10
0000 11 THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 TRANSFERRED.
0000 17
0000 18 THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 CORPORATION.
0000 21
0000 22 DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24
0000 25 *****
0000 26
0000 27 D. N. CUTLER 9-JAN-78
0000 28
0000 29 MODIFIED BY:
0000 30
0000 31
0000 32 V03-003 TMK0001 Todd M. Katz 05-Apr-1984
0000 33 Create and delete the logical name SYSS$PUTMSG using the
0000 34 new logical name system services $CRELNM and $DELLNM instead
0000 35 of the old system services $CRELOG and $DELLOG.
0000 36
0000 37 V03-002 DWT0065 David W. Thiel 18-Jan-1983
0000 38 Prevent creation of files named SYSS$ERROR and SYSS$OUTPUT.
0000 39
0000 40 V03-001 KDM0002 Kathleen D. Morse 28-Jun-1982
0000 41 Added $$$DEF.
0000 42
0000 43 **
```

```
0000 45 .SBTTL DATA DEFINITIONS
0000 46
0000 47 :
0000 48 : MACRO LIBRARY CALLS:
0000 49 :
0000 50
0000 51 $FABDEF ;DEFINE FAB OFFSETS
0000 52 $LNMDEF ;DEFINE LOG NAME SYSTEM SERVICE OFFSETS
0000 53 $PSLDEF ;DEFINE PROCESSOR STATUS FIELDS
0000 54 $RABDEF ;DEFINE RAB OFFSETS
0000 55 $SSDEF ;DEFINE SYSTEM STATUS CODES
0000 56
0000 57 :
0000 58 : LOCAL SYMBOLS:
0000 59 :
0000 60
00000100 0000 61 BUFSIZ=256 ;MAXIMUM RECORD SIZE
0000011B 0000 62 ESCAPE=108+27 ;ESCAPE EQUIVALENCE NAME PREFIX
0000 63
0000 64 :
0000 65 : LOCAL DATA:
0000 66 :
0000 67
00000000 0000 68 .PSECT YEXEPAGED2,BYTE
0000 69
3A 52 4F 52 52 45 24 53 59 53 00' 0000 70 ERRNAM: .ASCIC /SYSS$ERROR:/ ;'SYSS$ERROR' FILE NAME STRING
0A 0000
000B 71
47 4F 4C 2E 00' 000B 72 FILTYP: .ASCIC /.LOG/ ;DEFAULT FILE NAME STRING
04 000B
0010 73
47 53 4D 54 55 50 24 53 59 53 00' 0010 74 ISINAM: .ASCIC /SYSS$PUTMSG/ ;'SYSS$PUTMSG' FILE NAME STRING
0A 0010
3A 54 55 50 54 55 4F 24 53 59 53 00' 001B 75
0B 001B 76 OUTNAM: .ASCIC /SYSS$OUTPUT:/ ;'SYSS$OUTPUT' FILE NAME STRING
0027 77
52 50 24 4D 4E 4C 0000002F'010E0000' 0027 78 TABLE_NAME:
53 53 45 43 4F 0035 79 .ASCID /LNMS$PROCESS/ ;DESCRIPTOR FOR LNMS$PROCESS
```

```
003A 81 .SBTTL OPEN MESSAGE FILES FOR SYSTEM OUTPUT AND ERROR
003A 82 :+
003A 83 EXESOPEN_MSG - OPEN FILES FOR SYSTEM OUTPUT AND ERROR
003A 84 :
003A 85 THIS ROUTINE IS CALLED TO OPEN FILES FOR SYSTEM OUTPUT AND ERROR.
003A 86 :
003A 87 INPUTS:
003A 88 :
003A 89 NONE.
003A 90 :
003A 91 OUTPUTS:
003A 92 :
003A 93 RAB'S ARE ALLOCATED ON THE STACK FOR THE FILES 'SYS$OUTPUT' AND
003A 94 'SYS$ERROR', AND THE FILE ISI/IFI VALUES ARE OBTAINED EITHER BY ACTUALLY
003A 95 OPENING THE RESPECTIVE FILES OR BY LOGICAL NAME TRANSLATION.
003A 96 :
003A 97 R10 = ADDRESS OF ALLOCATED 'SYS$OUTPUT' RAB.
003A 98 R11 = ADDRESS OF ALLOCATED 'SYS$ERROR' RAB.
003A 99 :-
003A 100 :
003A 101 EXESOPEN_MSG:: ;OPEN MESSAGE FILES FOR SYSTEM OUTPUT AND ER
003A 102 :
003A 103 :
003A 104 ALLOCATE 2 RABS FOR SYS$OUTPUT AND SYS$ERROR ON THE CALLER'S STACK
003A 105 :
003A 106 :
003A 107 POPL R0 ;SAVE RETURN ADDRESS
003D 108 MOVAB -RAB$C_BLN(SP),R10 ;ALLOCATE SPACE FOR 'SYS$OUTPUT' RAB
0041 109 MOVAB -2*RAB$C_BLN(SP),R11 ;ALLOCATE SPACE FOR 'SYS$ERROR' RAB
0046 110 MOVL R11,SP ;SET NEW STACK ADDRESS
0049 111 PUSHL R0 ;RESTORE RETURN ADDRESS
004B 112 :
004B 113 :
004B 114 ALLOCATE AND INITIALIZE A FAB FOR OPENING THE FILES
004B 115 :
004B 116 :
004B 117 MOVAB -FAB$C_BLN(SP),SP ;ALLOCATE SPACE FOR FAB
004F 118 MOVL SP,R9 ;SAVE ADDRESS OF FAB
0052 119 MOVCS #0,(R9),#0,#FAB$C_BLN,(R9) ;CLEAR FAB
005A 120 MOVAB #FAB$C_BLN,FAB$B_BLN(R9) ;SET LENGTH OF FAB
005F 121 MOVAB #FAB$C_BID,FAB$B_BID(R9) ;SET IDENTIFICATION OF FAB
0062 122 MOVAB FILTYP,FAB$B_ONSTR(R9) ;SET DEFAULT NAME STRING SIZE
0067 123 MOVAB FILTYP+1,FAB$L_DNA(R9) ;SET DEFAULT NAME STRING ADDRESS
006C 124 MOVW #BUFSIZ,FAB$W_MRS(R9) ;SET MAXIMUM RECORD SIZE
0072 125 MOVAB #FAB$M_PUT,FAB$B_FAC(R9) ;SET FILE ACCESS MODE
0076 126 ASSUME FAB$M_SQO LE 255
0076 127 MOVL #FAB$M_CIF!FAB$M_SQO,FAB$L_FOP(R9) ;SET FILE OPEN OPTIONS
007E 128 MOVAB #FAB$M_CR,FAB$B_RAT(R9) ;SET ATTRIBUTES
0082 129 :
0082 130 :
0082 131 INITIALIZE BOTH RABS
0082 132 :
0082 133 :
0082 134 MOVCS #0,(R10),#0,#RAB$C_BLN,(R10) ;CLEAR OUTPUT FILE RAB
008A 135 MOVAB #RAB$C_BID,RAB$B_BID(R10) ;SET RAB BLOCK IDENTIFICATION
008D 136 MOVAB #RAB$C_BLN,RAB$B_BLN(R10) ;SET RAB BLOCK LENGTH
0092 137 MOVL R9,RAB$L_FAB(R10) ;SET ADDRESS OF FAB
```

69 0050 8F 00 6A 00 2C 0082 134  
01 A9 50 8F 90 008A 135  
35 A9 A6 AF 90 008D 136  
30 A9 A2 AF 90 0092 137  
36 A9 0100 8F B0 0092 137  
16 A9 01 90 0092 137  
04 A9 02000040 8F D0 0092 137  
1E A9 02 90 0092 137

```

0096 138 ASSUME RAB$V_EOF EQ 8
0096 139 MOVB #RAB$V_EOF-8,RAB$L_ROP+1(R10) ;APPEND TO END OF FILE
6B 05 AA 01 90 009A 140 MOVC #RAB$C_BLN,(R10),(R11) ;DUPLICATE OUTPUT FILE RAB
6A 0044 8F 28 00A0 141 ASSUME RAB$V_CCO EQ 31
07 AB 80 8F 90 00A0 142 MOVB #RAB$V_CCO-24,RAB$L_ROP+3(R11) ;SET CANCEL CONTROL 0
00A5 143
00A5 144
00A5 145 : TRANSLATE THE LOGICAL NAME SYS$PUTMSG TO SEE IF THE FILE IS ALREADY
00A5 146 : OPENED VIA A PREVIOUS CALL. IF THE LOGICAL NAME EXISTS, ITS TRANSLATION
00A5 147 : CONSISTS OF THE ISI'S FOR THE OUTPUT AND ERROR RAB, SO RE-ESTABLISH
00A5 148 : THE EXISTING STREAM AND BYPASS THE OPENS.
00A5 149
00A5 150
00A5 151 PUSHAB ISINAM+1 ;BUILD ISI NAME STRING DESCRIPTOR
7E FF68 CF 9F 00A9 152 MOVZBL ISINAM,-(SP)
FF63 CF 9A 00AE 153 MOVBL SP,R8 ;SAVE ADDRESS OF ISI NAME STRING DESCRIPTOR
58 5E D0 00B1 154 SUBL #12,SP ;ALLOCATE SPACE FOR EQUIVALENCE NAME
5E 0C C2 00B4 155 MOVL SP,R7 ;SAVE ADDRESS OF EQUIVALENCE NAME
57 5E D0 00B7 156 PUSHL SP ;BUILD EQUIVALENCE NAME DESCRIPTOR
DD 00B9 157 PUSHL #10
56 5E D0 00BB 158 MOVL SP,R6 ;SAVE ADDRESS OF EQUIVALENCE NAME DESCRIPTOR
00BE 159 $TRNLOG S LOGNAM=(R8),- ;TRANSLATE LOGICAL NAME
00BE 160 RSLBUF=(R6)
50 01 B1 00D1 161 CMPW S^#SS$_NORMAL,R0 ;NORMAL COMPLETION?
07 12 00D4 162 BNEQ 10$ ;IF NEQ NO
67 011B 8F B1 00D6 163 CMPW #ESCAPE,(R7) ;SPECIAL ESCAPE EQUIVALENCE NAME?
67 13 00DB 164 BEQL 40$ ;IF EQL YES
00DD 165
00DD 166
00DD 167 : CREATE FILES FOR SYS$OUTPUT AND SYS$ERROR AND SAVE RESULTANT ISI VALUES
00DD 168
00DD 169
00DD 170 10$: MOVAB ERRNAM,R0 ;GET ADDRESS OF 'SYS$ERROR' NAME STRING
50 FF1F CF 9E 00DD 171 BSBB CREATE_FILE ;CREATE 'SYS$ERROR' FILE
55 54 D0 00E2 172 MOVL R4,R5 ;SAVE 'SYS$ERROR' IFI
50 FF30 CF 9E 00E4 173 MOVAB OUTNAM,R0 ;GET ADDRESS OF 'SYS$OUTPUT' NAME STRING
73 10 00EC 174 BSBB CREATE_FILE ;CREATE 'SYS$OUTPUT' FILE
52 55 D0 00EE 175 MOVL R5,R2 ;SET 'SYS$ERROR' IFI
53 5B D0 00F1 176 MOVL R11,R3 ;SET 'SYS$ERROR' RAB
5D 10 00F4 177 BSBB CONNECT_RAB ;CONNECT RECORD STREAM
55 54 B1 00F6 178 CMPW R4,R5 ;'SYS$OUTPUT' AND 'SYS$ERROR' IFI'S MATCH?
0A 13 00F9 179 BEQL 20$ ;IF EQL YES
00FB 180
00FB 181 : 'SYS$ERROR' AND 'SYS$OUTPUT' ARE NOT THE SAME FILE
00FB 182
00FB 183
00FB 184
52 54 D0 00FB 185 MOVL R4,R2 ;SET 'SYS$OUTPUT' IFI
53 5A D0 00FE 186 MOVL R10,R3 ;SET 'SYS$OUTPUT' RAB
50 10 0101 187 BSBB CONNECT_RAB ;CONNECT RECORD STREAM
05 11 0103 188 BRB 30$
0105 189
0105 190 : 'SYS$ERROR' AND 'SYS$OUTPUT' ARE SAME FILE
0105 191
0105 192
0105 193
02 AA 02 AB B0 0105 194 20$: MOVW RAB$W_ISI(R11),RAB$W_ISI(R10) ;COPY ISI NUMBER
```

```
010A 195
010A 196 :
010A 197 : CREATE LOGICAL NAME "SYSS$PUTMSG" EQUIVALENCED TO ERROR AND OUTPUT ISI NUMBERS
010A 198 :
010A 199 :
02 67 011B 8F B0 010A 200 30$: MOVW #ESCAPE,(R7) ;INSERT SPECIAL ESCAPE PREFIX
04 A7 02 AA B0 010F 201 MOVW RAB$W_ISI(R10),2(R7) ;INSERT 'SYSS$OUTPUT' ISI NUMBER
06 A7 02 AB B0 0114 202 MOVW RAB$W_ISI(R11),4(R7) ;INSERT 'SYSS$ERROR' ISI NUMBER
08 A7 54 B0 0119 203 MOVW R4,6(R7) ;INSERT 'SYSS$OUTPUT' IFI NUMBER
08 A7 55 B0 011D 204 MOVW R5,8(R7) ;INSERT 'SYSS$ERROR' IFI NUMBER
0121 205
7E 7C 0121 206 CLRQ -(SP) ;ZERO END OF ITEM LIST MARKER
0123 207 ;AND STRING RESULTANT LENGTH BUFFER
0002000A 67 9F 0123 208 PUSHAB (R7) ;ADDRESS OF TRANSLATION STRING
51 5E D0 0125 209 PUSHL #<LNMS_STRING @ 16 + 10> ;STRING ITEM TYPE AND STRING BUFFER SIZE
012B 210 MOVL SP,R1 ;ADDRESS OF ITEM LIST
012E 211
012E 212 $CRELNM_S - ;CREATE THE LOGICAL NAME SYSS$PUTMSG
012E 213 ITMLST = (R1),- ;ADDRESS OF ITEM LIST
012E 214 LOGNAM = (R8),- ;ADDRESS OF LOGICAL NAME DESCRIPTOR
012E 215 TABNAM = TABLE_NAME ;ADDRESS OF TABLE NAME DESCRIPTOR
5E 10 C0 0141 216 ADDL2 #16,SP ;REMOVE ITEM LIST FROM STACK
0144 217
0144 218 :
0144 219 : INSERT ISI NUMBERS IN RABS AND RETRIEVE ERROR/OUTPUT EQUIVALENCE INDICATOR
0144 220 :
0144 221 :
02 AA 02 A7 B0 0144 222 40$: MOVW 2(R7),RAB$W_ISI(R10) ;INSERT 'SYSS$OUTPUT' ISI NUMBER
02 AB 04 A7 B0 0149 223 MOVW 4(R7),RAB$W_ISI(R11) ;INSERT 'SYSS$ERROR' ISI NUMBER
5E 6C AE 9E 014E 224 MOVAB FAB$C_BLN+8*12+8(SP),SP ;REMOVE ALLOCATED SPACE FROM STACK
05 0152 225 RSB ;
```

```

0153 227 .SBTTL INTERNAL SUBROUTINES
0153 228
0153 229
0153 230 : CONNECT_RAB - CONNECT RECORD STREAM
0153 231 :
0153 232 : THIS ROUTINE IS CALLED TO CONNECT A RECORD STREAM TO A RAB.
0153 233 :
0153 234 : INPUTS:
0153 235 :
0153 236 : R2 = FILE IFI NUMBER.
0153 237 : R3 = ADDRESS OF RAB.
0153 238 :
0153 239 : OUTPUTS:
0153 240 :
0153 241 : THE SPECIFIED RAB IS CONNECTED AND IMPLIED CARRIAGE CONTROL IS INSERTED
0153 242 : IN THE RESULTANT ISI NUMBER.
0153 243 :
0153 244 : NOTE ALL ERRORS ARE IGNORED.
0153 245 :
0153 246
0153 247 CONNECT_RAB:
02 A9 52 B0 0153 248 MOVW R2,FAB$W IFI(R9) ;CONNECT RAB
0157 249 $CONNECT RAB=(R3) ;INSERT IFI NUMBER
05 0160 250 RSB ;CONNECT RECORD STREAM
;

```

```
0161 252
0161 253
0161 254 : CREATE_FILE
0161 255
0161 256 : THIS ROUTINE IS CALLED TO CREATE A FILE AND RETURN THE RESULTANT IFI.
0161 257
0161 258 : NOTE THAT WHATEVER AMPLIFIED PRIVILEGES THE IMAGE MAY HAVE ARE TEMPORARILY
0161 259 : REMOVED BY THIS ROUTINE DURING ITS EXECUTION. THIS IS TO PREVENT A USER
0161 260 : FROM WRITING BOGUS FILES BY REDEFINING SYSS$OUTPUT OR SYSS$ERROR.
0161 261
0161 262 : INPUTS:
0161 263
0161 264 : R0 = ADDRESS OF FILE NAME COUNTED STRING.
0161 265
0161 266 : OUTPUTS:
0161 267
0161 268 : R4 = RESULTANT FILE IFI NUMBER.
0161 269
0161 270 : NOTE ALL ERRORS ARE IGNORED.
0161 271
0161 272
0161 273 CREATE_FILE:
0161 274 :CREATE FILE
2C A9 80 90 0161 274 MOVB (R0)+,FAB$B_FNS(R9) :INSERT FILENAME STRING SIZE
0165 275 MOVAB (R0),FAB$L_FNA(R9) :INSERT FILE NAME ADDRESS
0169 276
0169 277 PUSH R2 :SAVE R2 AND ALLOCATE PRIVILEGE MASK BUFFER
52 07 BB 0169 277 MOVL SP,R2 :POINT TO PRIVILEGE MASK BUFFER
016B 278 $SETPRV_S PRMFLG=#1,PRVPRV=(R2) :READ PROCESS CURRENT PRIVILEGES
016E 279 MCOML (R2),(R2) :GENERATE COMPLEMENT
04 A2 04 A2 D2 017D 280 MCOML 4(R2),4(R2) :OF PRIVILEGE MASK
0185 281 $SETPRV_S PRMFLG=#0,ENBFLG=#0,PRVADR=(R2),PRVPRV=(R2) :CLEAR AMPLIFIED PRIVI
0194 282
0194 283
0194 284 CLRW FAB$W_IFI(R9) :CLEAR IFI NUMBER
1E A9 02 90 0197 285 MOVB #FAB$M_CR,FAB$B_RAT(R9) :SET RECORD ATTRIBUTES
1F A9 02 90 019B 286 MOVB #FAB$C_VAR,FAB$B_RFM(R9) :SET RECORD FORMAT
019F 287 MOVPSL R0 :READ CURRENT PSL
02 18 ED 01A1 288 CMPZV #PSL$V_CURMOD,#PSL$S_CURMOD,- :CURRENT MODE USER?
03 50 01A4 289 R0,#PSL$C_USER
00 04 A9 12 E2 01A6 290 BEQL 10$ :IF EQL YES
54 02 A9 3C 01A8 291 BBSS #FAB$V_PPF,FAB$L_FOP(R9),10$ :SET PROCESS PERMANENT FILE
01AD 292 $CREATE FAB=(R9) :CREATE FILE
01B6 293 MOVZWL FAB$W_IFI(R9),R4 :GET RESULTANT IFI NUMBER
01BA 294
01BA 295 $SETPRV_S PRMFLG=#0,ENBFLG=#1,PRVADR=(R2) :RESTORE IMAGE PRIVILEGES
07 BA 01C9 296 POPR #M<R0,R1,R2> :CLEAN STACK AND RESTORE R2
05 01CB 297 RSB
```

```

01CC 299 .SBTTL CLOSE MESSAGE FILES
01CC 300
01CC 301 :+
01CC 302 : EXE$CLOSE_MSG - CLOSE MESSAGE FILES OPENED BY EXE$OPEN_MSG
01CC 303
01CC 304 : THIS ROUTINE IS CALLED TO CLOSE THE FILES FOR USED TO OUTPUT
01CC 305 : MESSAGES. IF CALLER IN USER MODE, THE FILES ARE LEFT OPEN AS
01CC 306 : AN OPTIMIZATION SINCE THEY WILL EVENTUALLY GET CLOSED BY IMAGE
01CC 307 : RUNDOWN. HOWEVER, FOR OTHER ACCESS MODES, THE FILES MUST BE
01CC 308 : CLOSED HERE.
01CC 309
01CC 310 : INPUTS:
01CC 311
01CC 312 : LOGICAL NAME "SYSS$PUTMSG" CONTAINING 5 WORDS:
01CC 313
01CC 314 : ESCAPE,OUTISI,ERRISI,OUTIFI,ERRIFI
01CC 315
01CC 316 : OUTPUTS:
01CC 317
01CC 318 : THE FILES ARE CLOSED, IF POSSIBLE.
01CC 319 :-
01CC 320
01CC 321 EXE$CLOSE_MSG::
01CC 322
01CC 323 :
01CC 324 : IF WE ARE IN USER MODE, KEEP THE FILES OPEN AS AN OPTIMIZATION
01CC 325 : BECAUSE THEY WILL EVENTUALLY GET CLOSED BY RMS RUNDOWN WHEN THE
01CC 326 : IMAGE IS RUNDOWN.
01CC 327 :
01CC 328
01CC 329 MOVPSL R0 ;GET CURRENT PSL
01CE 330 CMPZV #PSL$V_CURMOD,#PSL$S_CURMOD,R0,#PSL$C_USER ;IN USER MODE?
01D3 331 BNEQ 10$ ;BRANCH IF NOT
01D5 332 RSB ;EXIT - LEAVE FILES OPEN
01D6 333
01D6 334 :
01D6 335 : ALLOCATE AND INITIALIZE A FAB FOR OPENING THE FILES
01D6 336 :
01D6 337
01D6 338 10$: PUSHR #^M<R2,R3,R4,R5>
01D8 339 MOVAB -FAB$C_BLN(SP),SP ;ALLOCATE SPACE FOR FAB
01DC 340 MOVCS #0,(SPT,#0,#FAB$C_BLN,(SP) ;CLEAR FAB
01E4 341 MOVL SP,R2 ;SAVE ADDRESS OF FAB
01E7 342 ASSUME FAB$B_BLN EQ FAB$B_BID+1
01E7 343 MOVW #FAB$C_BID+<FAB$C_BLN*8>,FAB$B_BID(R2) ;SET BLOCK ID/LENGTH
01EC 344
01EC 345 :
01EC 346 : TRANSLATE THE LOGICAL NAME SYSS$PUTMSG TO SEE IF THE FILE IS OPENED VIA
01EC 347 : EXE$OPEN_MSG. IF THE LOGICAL NAME EXISTS, GET THE IFI'S AND USE THEM
01EC 348 : TO CLOSE THE FILES.
01EC 349 :
01EC 350
01EC 351 PUSHAB ISINAM+1 ;BUILD ISI NAME STRING DESCRIPTOR
01F0 352 MOVZBL ISINAM,-(SP)
01F5 353 MOVL SP,R3 ;SAVE ADDRESS OF ISI NAME STRING DESCRIPTOR
01F8 354 SUBL #12,SP ;ALLOCATE STORAGE FOR EQUIVALENCE NAME
01FB 355 MOVL SP,R4 ;SAVE ADDRESS OF EQUIVALENCE NAME

```

```
50 5E DD 01FE 356      PUSHL SP      ;BUILD EQUIVALENCE NAME DESCRIPTOR
    OA DD 0200 357      PUSHL #10
    5E DD 0202 358      MOVL SP,R0      ;SAVE ADDRESS OF EQUIVALENCE NAME DESCRIPTOR
    0205 359      $TRNLOG,S LOGNAM=(R3),- ;TRANSLATE LOGICAL NAME
    0205 360      RSLBUF=(R0)
    50 01 B1 0218 361      CMPW S^#SS$_NORMAL,R0 ;NORMAL COMPLETION?
    39 12 021B 362      BNEQ 50$      ;IF NEQ NO
    64 011B 8F B1 021D 363      CMPW #ESCAPE,(R4) ;SPECIAL ESCAPE EQUIVALENCE NAME?
    32 12 0222 364      BNEQ 50$      ;IF EQL YES
    0224 365
    0224 366
    0224 367      : CLOSE SYS$ERROR
    0224 368
    0224 369
    02 A2 06 A4 B0 0224 370      MOVW 6(R4),FAB$W_IFI(R2) ;SET IFI INTO FAB
    0229 371      $CLOSE FAB=(R2) ;CLOSE SYS$ERROR
    0232 372
    0232 373
    0232 374      : IF SYS$OUTPUT IS NOT THE SAME AS SYS$ERROR, THEN CLOSE IT
    0232 375
    08 A4 06 A4 B1 0232 376      CMPW 6(R4),8(R4) ;BOTH IFI'S THE SAME?
    OE 13 0237 377      BEQL 20$ ;BRANCH IF SO
    02 A2 08 A4 B0 0239 378      MOVW 8(R4),FAB$W_IFI(R2) ;SET IFI INTO FAB
    023E 379      $CLOSE FAB=(R2) ;CLOSE SYS$OUTPUT
    0247 380
    0247 381
    0247 382      : DELETE LOGICAL NAME 'SYS$PUTMSG' TO INDICATE FILES CLOSED.
    0247 383
    0247 384
    0247 385 20$: $DELLNM,S - ;DELETE LOGICAL NAME FOR SYS$PUTMSG
    0247 386      LOGNAM = (R3),- ;ADDRESS OF LOGICAL NAME DESCRIPTOR
    0247 387      TABNAM = TABLE_NAME ;ADDRESS OF TABLE NAME DESCRIPTOR
    0256 388
    5E 6C AE 9E 0256 389 50$: MOVAB FAB$C,BLN+8+12+8(SP),SP ;DEALLOCATE FAB AND SCRATCH SPACE
    3C BA 025A 390      POPR #^M<R2,R3,R4,R5> ;RESTORE REGISTERS
    05 025C 391      RSB
    025D 392      .END
```

OPENMSG  
Symbol table

E 1

- OPEN MESSAGE FILES FOR SYSTEM OUTPUT A 16-SEP-1984 00:39:27 VAX/VMS Macro V04-00  
5-SEP-1984 03:45:38 [SYS.SRC]OPENMSG.MAR;1

Page 10  
(6)

\$\$TMP1	=	00000001		
\$\$TMP2	=	00000062		
BUFSIZ	=	00000100		
CONNECT_RAB		00000153	R	02
CREATE_FILE		00000161	R	02
ERRNAM		00000000	R	02
ESCAPE	=	0000011B		
EXESCLOSE MSG		000001CC	RG	02
EXE\$OPEN MSG		0000003A	RG	02
FAB\$B_BID	=	00000000		
FAB\$B_BLN	=	00000001		
FAB\$B_DNS	=	00000035		
FAB\$B_FAC	=	00000016		
FAB\$B_FNS	=	00000034		
FAB\$B_RAT	=	0000001E		
FAB\$B_RFM	=	0000001F		
FAB\$C_BID	=	00000003		
FAB\$C_BLN	=	00000050		
FAB\$C_VAR	=	00000002		
FAB\$L_DNA	=	00000030		
FAB\$L_FNA	=	0000002C		
FAB\$L_FOP	=	00000004		
FAB\$M_CIF	=	02000000		
FAB\$M_CR	=	00000002		
FAB\$M_PUT	=	00000001		
FAB\$M_SQO	=	00000040		
FAB\$V_PPF	=	00000012		
FAB\$W_IFI	=	00000002		
FAB\$W_MRS	=	00000036		
FILTYP		0000000B	R	02
ISINAM		00000010	R	02
LNMS_STRING	=	00000002		
OUTNAM		0000001B	R	02
PSL\$C_USER	=	00000003		
PSL\$S_CURMOD	=	00000002		
PSL\$V_CURMOD	=	00000018		
RAB\$B_BID	=	00000000		
RAB\$B_BLN	=	00000001		
RAB\$C_BID	=	00000001		
RAB\$C_BLN	=	00000044		
RAB\$L_FAB	=	0000003C		
RAB\$L_ROP	=	00000004		
RAB\$M_CCO	=	80000000		
RAB\$M_EOF	=	00000100		
RAB\$V_CCO	=	0000001F		
RAB\$V_EOF	=	00000008		
RAB\$W_ISI	=	00000002		
SSS_NORMAL	=	00000001		
SYSSCLOSE	*****		GX	02
SYSSCONNECT	*****		GX	02
SYSSCREATE	*****		GX	02
SYSSCRELNM	*****		GX	02
SYSSDELLNM	*****		GX	02
SYSSSETPRV	*****		GX	02
SYSSTRNLOG	*****		GX	02
TABLE_NAME		00000027	R	02

+-----+  
! Psect synopsis !  
+-----+

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 ( 0.)	00 ( 0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$AB\$\$	00000000 ( 0.)	01 ( 1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
YEXEPAGED2	0000025D ( 605.)	02 ( 2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

+-----+  
! Performance indicators !  
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.07	00:00:00.83
Command processing	106	00:00:00.54	00:00:05.03
Pass 1	271	00:00:08.23	00:00:25.49
Symbol table sort	0	00:00:01.10	00:00:02.86
Pass 2	82	00:00:01.67	00:00:05.76
Symbol table output	8	00:00:00.08	00:00:00.08
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	500	00:00:11.72	00:00:40.08

The working set limit was 1350 pages.

46490 bytes (91 pages) of virtual memory were used to buffer the intermediate code.

There were 40 pages of symbol table space allocated to hold 799 non-local and 8 local symbols.

392 source lines were read in Pass 1, producing 14 object records in Pass 2.

23 pages of virtual memory were used to define 21 macros.

+-----+  
! Macro library statistics !  
+-----+

Macro library name	Macros defined
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	0
-\$255\$DUA28:[SYS.LIB]STARLET.MLB;2	18
TOTALS (all libraries)	18

959 GETS were required to define 18 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:OPENMSG/OBJ=OBJ\$:OPENMSG MSRC\$:OPENMSG/UPDATE=(ENH\$:OPENMSG)+EXECMLS/LIB

0377

**DIGITAL  
CONFIDE**

EQUIPMENT  
INITIAL AND

CORPORATION  
PROPRIETARY

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY